An Introduction to National's SPI Interface EEPROMs

National Semiconductor
Application Note 832
Robert Stodieck
July 1992

Abstract: National's new NM25C04 serial EEPROM can write from 1 to 4 bytes at a time and has a sequential read capability. The command protocol is straight forward and byte-oriented. The result is a serial EEPROM to be taken seriously for any application not using a microcontroller with a dedicated MICROWIRE™ interface

What is SPI?

The Serial Peripheral Interface (SPI) is a general purpose synchronous serial interface originally found on certain Motorola microcontrollers. A virtually identical interface can now be found on certain TI and SGS Thompsen microcontrollers as well. The SPI should not be confused with the SCI (Serial Communications Interface) frequently also found on the same microcontrollers but usually used for asynchronous communications.

Since Many SPI Protocols are Possible, What Convention has been Adopted for the EEPROM?

Unlike the MICROWIRE and Inter-Integrated Circuit (IIC) serial buses, the Serial Peripheral Interface (SPI) does not completely define a data transfer protocol. National's 25Cxx SPI EEPROM interface convention assumes that the Motorola Microcontroller's SPI registers are set with:

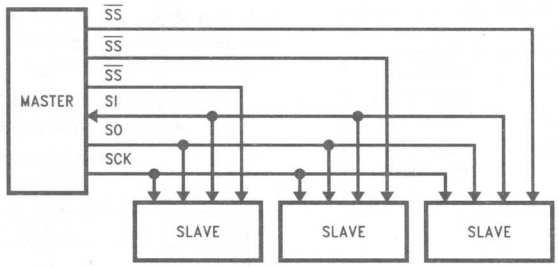
Clock Phase Bit = 1
Clock Polarity Bit = 0

These same settings are usable for the SPI interfaces found on other manufacturer's microcontrollers as well. In this data transfer convention, Slave Selects reset the serial logic between transfers. The Slave Selects are asserted low. Data transfer lengths are multiples of 8-bit bytes.

Reads and writes to the EEPROM are initiated by enabling the device by asserting the Slave Select pin low and shifting an opcode and address fields to the EEPROM *(Figure 2)*. This requires 16 bits of data or two full transfers from the 8-bit data register on the microcontrollers SPI. For write operations this is followed by one to four bytes of the data to be written. The write operation is internally self-timed and begins when the Slave Selects become low. For read operations the address is answered by bytes of data on the SO pin starting from the address requested and sequencing upward as many times as is desired, wrapping around at the end of the memory array.

NATIONAL'S NM25C04 EEPROM FEATURES AND OPERATION OVERVIEW

- 512 x 8-bits
- Write protect pin
- No need to pre-erase locations before write
- One to four byte block write—wraps at end of four byte memory blocks
- Unlimited sequential read—wraps at end of memory array
- Four zone software initialized write protection to prevent unintended overwrites to reserved areas
- Internally self-timed write cycles
- Write cycle ready/busy indication via polling internal status register—does not tie up the output bus during write timeout



TL/D/11473–1

FIGURE 1. The physical Serial Peripheral Interface definition is very similar to National's MICROWIRE. "Slave" Selects, Serial Clock, Slave In (SI), Slave Out (SO). The SPI interface can also be used to access MICROWIRE devices.

RRD-B20M92/Printed in U. S. A.

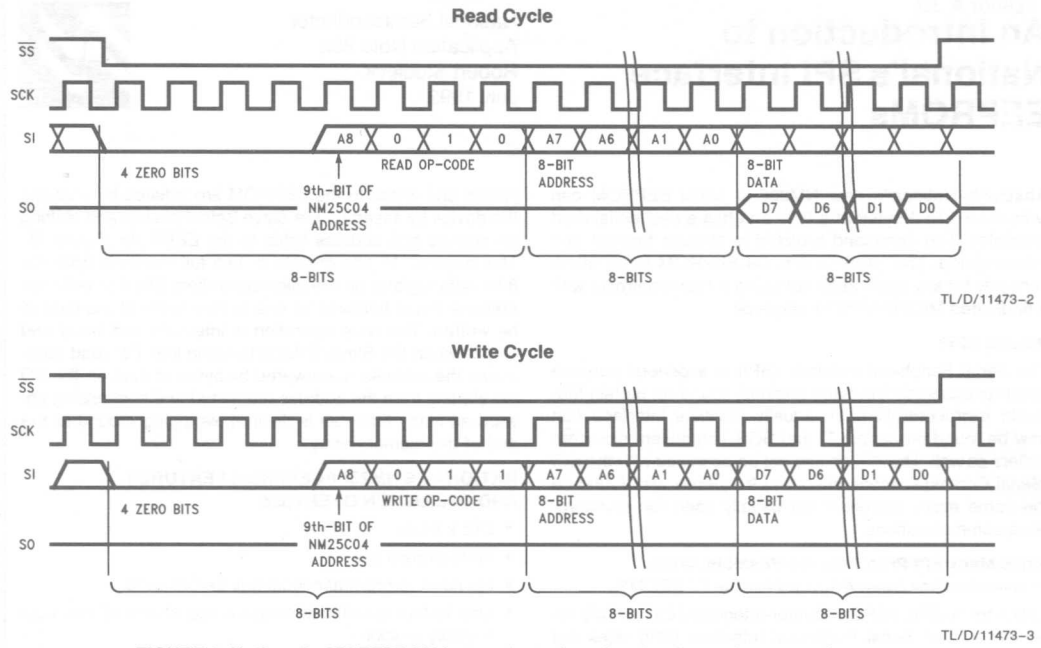
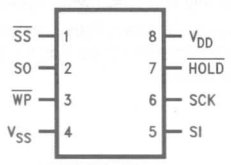


FIGURE 2. National's SPI EEPROM byte-oriented read and write cycle conventions. Note that the read cycle can be extended and data bytes from sequential bytes will be output. Write cycles can actually include 1 to 4 bytes to be written into a 1 to 4 location "block".

- Clock gate pin (hold)
- Slave In (SI)
- Slave Out (SO)
- Slave Select ($\overline{SS}$)
- Serial Clock (SCK)
- Serial Transfer Hold ($\overline{HOLD}$)
- Write Protect ($\overline{WP}$)



TL/D/11473-10

OPCODES

WREN	Write Enable (Note 1)	0000 X 110
WRDI	Write Disable	0000 X 100
RDSR	Read Status Register	0000 X 101
WRSR	Write Status Register	0000 X 001
READ	Read	0000 A8 011
WRITE	Write	0000 A8 010

Note 1: Part powers up in write disable mode.

WRITE COMPLETE INDICATION VIA STATUS REGISTER

All EEPROMs have write cycles that are far longer than normal read cycles. Since writes are always relatively infrequent, it is often possible to simply wait out this time period before proceeding with other operations. However, the read status register opcode provides a method of polling for the end of write indication that does not force the processor to wait before proceeding to another operation. In fact, the only time it is absolutely required that one check the write complete status is prior to another read or write attempt. Unlike most serial EEPROMs this is practical on the NM25C04 by the read status register (RDSR) operation (see Table I). The RDY bit in the status register is low if the EEPROM has completed all write operations and is ready for another read or write.

Note that Status register also provides the status of the zone protect bits BP1 and BP0, and reveals whether the device is currently enabled for writing the WEN bit.

TABLE I. Status Register Format and Zone Write Protection Ranges

Status Register Format

Bit	7	6	5	4	3	2	1	0
Value	1	1	1	1	BP1	BP0	WEN	$\overline{RDY}$

Write Protect Ranges

BP1	BP0	Write Protected Range
0	0	None
0	1	180–1FF
1	0	100–1FF
1	1	000–1FF

BLOCK WRITE AND SEQUENTIAL READ

One to four bytes can be written within a "block" boundary. The first write "block" is at addresses 0 to 3, the second is at addresses 4 to 7 etc. The block to be written to is loaded in one sequence by simply loading an additional byte after the first byte shown in the *Figure 2* write cycle example. If address "0" were requested and four bytes are sent, the first byte will go to address zero and the second would go to address two etc. If address "3" were requested and two bytes are sent, the first byte will go to address three and the second would wrap around the block boundary and be written to address zero. Block write allows four bytes at a time, thus reducing the average write time. This is sometimes important due to the very long inherent write cycle times of EEPROMs.

Sequential read allows successive bytes to be read out of the EEPROM without having to re-send a read opcode or address for each new byte. This feature allows significant improvements in the average read access time for multibyte data. The entire length of memory can be read in one operation if required. The read sequence will wrap back to location zero if read past the end of memory.

WRITE PROTECT FEATURES

EEPROMs operate on 5V supplies and do not require a separate high-voltage supply to execute write operations. This is a primary feature of EEPROMs but inherent in this capability are some unique problems as well. Since they can be programmed on board and data in memory is non-volatile, accidental overwrite is a constant possibility. Unlike the majority of components on a typical PC board, the non-volatile characteristic assures that the problem will not correct itself after reset or power down. These problems are far less common in serial I/O devices than in parallel EEPROMs but it remains a serious issue. Thus a number of write protect strategies can be supported on the NM25C04.

SOFTWARE WRITE ENABLE/DISABLE

All of National's Serial EEPROMs power up in a write disable state to prevent accidental writes in the noise of the power-up operation. In order to actually write to the device, a write enable instruction must be sent to the EEPROM before a write instruction can be successfully executed. This is the function of the **WREN** (WRite ENable) instruction. The write function can be disabled again with the **WRDI** (WRite DIsable) instruction. The requirement that one execute a write enable instruction before an actual write operation greatly reduces the probability of a random noise induced write over. It can also be used to manage software related problems or to assist in debug.

ZONE WRITE PROTECTION

The NM25C04 has the ability to disable writes via software to certain areas of EEPROM. This is done with the **WRSR** (WRite Status Register) instruction. The write protect ($\overline{WP}$) pin must be high to allow writes to the status register and the **WRSR** instruction must **always** be preceded by a WREN (WRite ENable) instruction. Execution of the WRSR instruction always places the device in the write disable state. Thus the WREN instruction **must** be executed before any further writes to the EEPROM can occur.

The values in two non-volatile bits of the status register control whether the higher ¼, the higher ½, or the entire array is rendered unwritable. This write protection feature is under software control but is non-volatile. Protected areas become read only memory. The **RDSR** (ReaD Status Register) instruction allows the state of the write disable bits and the write status bits to be read. The two non-volatile write protect bits are writable via the **WRSR** (WRite Status Register) instruction. They may be altered repeatedly.

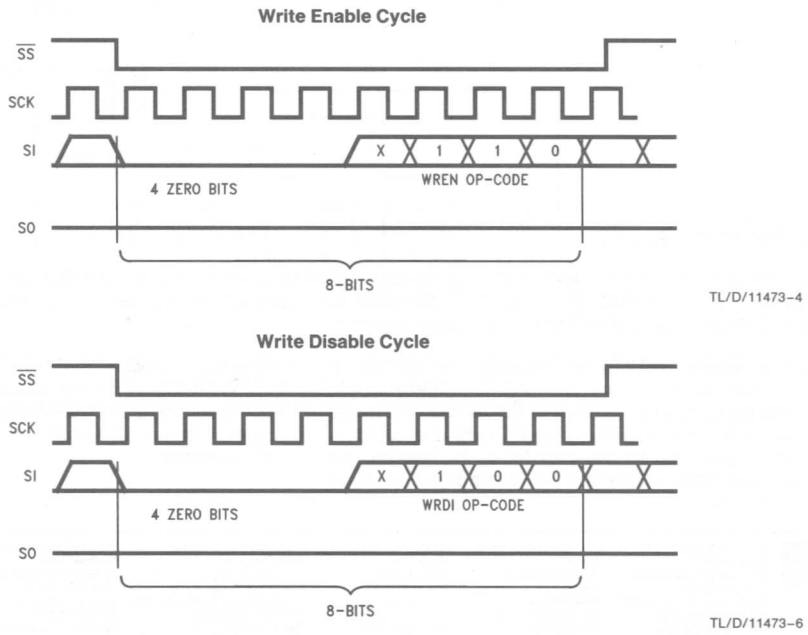


FIGURE 3. Read and Write Status Register Cycles

WRITE PROTECT PIN

The write protect pin provides a hardware method of preventing writes. The write protect pin must be high to allow normal writes to the EEPROM or to the status register. The uses of this pin are numerous. A manufacturer may use it to prevent a serial number, date of manufacture, security code or similar information from being altered intentionally or unintentionally in the field. Thus the EEPROM may be pre-programmed in a programmer prior to assembly. Once installed in the PC board with the write protect pin tied low, the data in the EEPROM is permanent. Calibration data can be made to be alterable only in the factory or by a field engineer by controlling the write protect pin with a jumper or internal switch.

One of the more subtle problems is the possibility of overwrite in a microcontroller "crash". Microcontrollers are easily upset by V_{CC} transients caused by such things as relay or electric motor operation, intermittent power switch operation or battery replacements, etc. V_{CC} transients can induce glitches on reset, interrupt or clock lines but the result is generally similar, that is, the microcontroller's program counter will be upset. The microcontroller will leave its normal program sequence and begin running at an arbitrary program memory location. One of the possible "landing" sites is the code associated with writing the serial EEPROM. Since microcontroller address spaces are often quite small this is a serious possibility, especially if the transients are a common occurrence. Badly designed reset circuits can aggravate or cause this problem. This type of event can also defeat software based write protect strategies. The write protect pin can be used to safeguard against this type of problem.

WRITE PROTECT PIN APPLICATIONS EXAMPLES

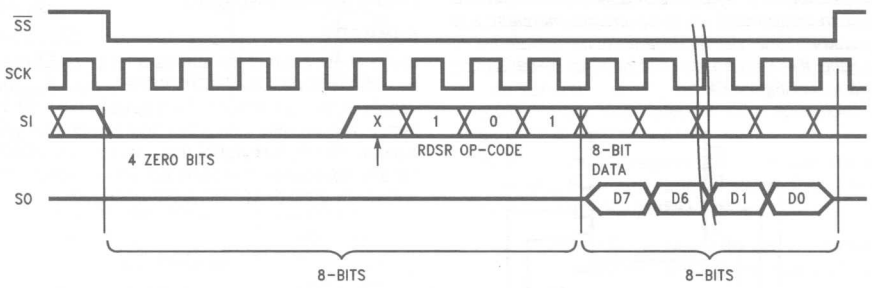
Oven or Refrigerator Temperature Controller *(Figure 5)*

A temperature controller uses an EEPROM to record both set point and calibration information. Use of the EEPROM to record the set temperature prevents loss of the set temperature if the unit is shut off. Calibration data for the temperature sensor is also recorded in EEPROM initially at the factory but may be recalibrated by a technician in the field. In the interests of compactness and economy the microcontroller is located in close proximity to an electromagnetic relay used to control a larger off-board power relay which actually switches power on or off to a heater or refrigeration unit. This is the only output required from this board.

In this design the power relay shares the same power cord as the digital logic board. This is the normal case for refrigerators and ovens. The power line is subject to numerous transients types not the least of which is caused by the routine turn-on and turn-off of the relays, heating elements or cooling systems of the appliance itself, but which also includes lightning, outages, imperfectly inserted cords and etc. Thus it is impossible to guarantee that the on-board microcontroller will run faultlessly throughout its service life. The problem is not that a destructive microcontroller malfunction is particularly probable, the problem is that such an event need only occur once to corrupt the data in the EEPROM and cause problems.

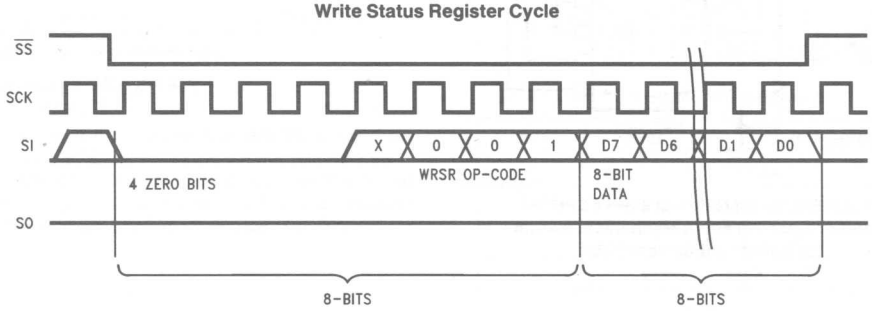
If the microcontroller "crashes" it is a simple matter to unplug the unit to reset the controller. In fact a simple watch dog timer can reset the microcontroller routinely so that such an event will not even be noticed. But it is possible that the calibration data or set point data in the EEPROM will be

Read Status Register Cycle



TL/D/11473–5

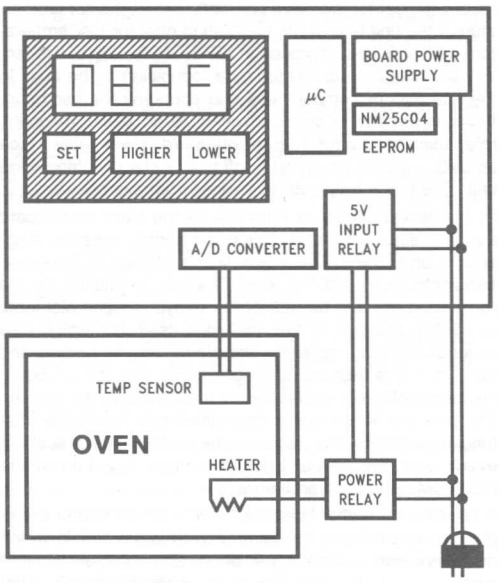
Write Status Register Cycle



TL/D/11473–7

FIGURE 4. Read and Write Status Register Cycles

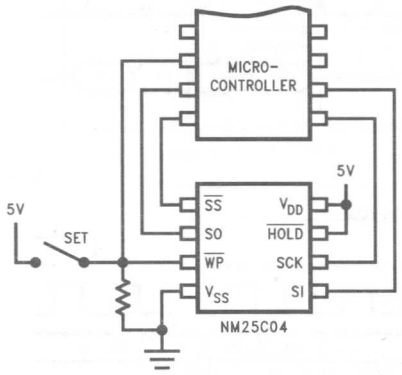
overwritten in such a crash. This may be true almost regardless of what software protection schemes are in place. And of course the non-volatile EEPROM will not "forget" the data from the transient event after powerdown or reset.



TL/D/11473–8

FIGURE 5. Simple Temperature Controller in an Oven Application

A fix to this problem is to hardwire the Write Protect pin to a switch or push button that must be closed manually any time the values in the EEPROM are to be updated. i.e., when changing the temperature set point or re-calibrating the controller (See Figure 6).



TL/D/11473–9

FIGURE 6. Using National's SPI EEPROM Write Protect Pin and a Mechanical Switch to Safeguard Critical Data

The more obvious approach of simply allowing the microcontroller to read the set switch and handle the write operation to the EEPROM should be avoided. This approach introduces the danger of subjecting the EEPROM to unintended writes from a temporarily wayward microcontroller.

Auto Cassette Deck Security Code Application

Auto Cassette Decks sometimes employ an anti-theft scheme requiring that the owner enter a security code into the unit from a keypad on the front panel before the unit can be used. In practice the actual security code is typically resident in a serial EEPROM. The initial code is entered at the factory and in some cases may be changed by the owner who has to know the original code in order to change the code.

The application problem has many similarities to the temperature controller example. It is important not to corrupt the data in the EEPROM. To do so would deny use of the cassette player to its owner. The power supply to the player is prone to transient events. The motors and control elements on the deck create V_{CC} transients and the car itself creates predictable power supply catastrophes. Cranking the car on a low battery, battery replacement and the inevitable jump start to name a few.

The fix is precisely the same as the temperature controller example. That is a mechanical switch can be used to enable writes, via the Write Protect pin, only at the time when a change is clearly intended by the user.

CLOCK GATING PIN ($\overline{\text{HOLD}}$)

For those applications where the EEPROM may be interfaced to relatively simple logic, a clock gating pin has been provided. Transitions of the $\overline{\text{HOLD}}$ pin must be restricted to times when the clock pin is high to avoid clock glitches. (This is a simple AND gate.)

SUMMARY

National established the standard for serial EEPROM devices with the NM93CXX MICROWIRE family. National offers a variety of devices and interface options. The SPI NM25CXX family devices offer designers new choices for varied applications. The feature set of this new family makes it suitable for any application that does not use a microcontroller with a dedicated IIC or MICROWIRE interface.